2 A simple example: WordPress in five minutes

This chapter covers

- Creating a blogging infrastructure
- Analyzing costs of a blogging infrastructure
- Exploring a blogging infrastructure
- Shutting down a blogging infrastructure

Having looked at why AWS is such a great choice to run web applications in the cloud, in this chapter, you'll evaluate migrating a simple web application to AWS by setting up a sample cloud infrastructure within five minutes.

> **NOTE** The example in this chapter is totally covered by the Free Tier (see section 1.4.1 for details). As long as you don't run this example longer than a few days, you won't pay anything for it. Keep in mind that this applies only if you created a fresh AWS account for this book and there is nothing else going on in your AWS account. Try to complete the chapter within a few days, because you'll clean up your account at the end of the chapter.

Imagine you work for a mid-sized company that runs a blog to attract new software and operations engineers. WordPress is used as the content management system.

Around 1,000 people visit the blog daily. You are paying $150 USD per month for the on-premises infrastructure. This seems expensive to you, particularly because at the moment the blog is suffering from several outages per month.

To leave a good impression on potential candidates, the infrastructure should be highly available, which is defined as an uptime of 99.99%. Therefore, you are evaluating new options to operate WordPress reliably. AWS seems to be a good fit. As a proof-of-concept, you want to evaluate whether a migration is possible. To do so, you need to do the following:

- Set up a highly available infrastructure for WordPress.
- Estimate monthly costs of the infrastructure.
- Come to a decision and delete the infrastructure afterward.

WordPress is written in PHP and uses a MySQL database to store data. Apache is used as the web server to serve the pages. With this information in mind, it's time to map your requirements to AWS services.

2.1 Creating your infrastructure

You'll use five different AWS services to copy the old infrastructure to AWS:

- *Elastic Load Balancing (ELB)*—AWS offers a load balancer as a service. The load balancer distributes traffic to a bunch of virtual machines, and is highly available by default. Requests are routed to virtual machines as long as their health check succeeds. You'll use the Application Load Balancer (ALB) which operates on Layer 7 (HTTP and HTTPS).
- *Elastic Compute Cloud (EC2)*—The EC2 service provides virtual machines. You'll use a Linux machine with an optimized distribution called Amazon Linux to install Apache, PHP, and WordPress. You aren't limited to Amazon Linux; you could also choose Ubuntu, Debian, Red Hat, or Windows. Virtual machines can fail, so you need at least two of them. The load balancer will distribute the traffic between them. In case a virtual machine fails, the load balancer will stop sending traffic to the failed VM, and the remaining VM will need to handle all requests until the failed VM is replaced.
- *Relational Database Service (RDS) for MySQL*—WordPress relies on the popular MySQL database. AWS provides MySQL with its RDS. You choose the database size (storage, CPU, RAM), and RDS takes over operating tasks like creating backups and installing patches and updates. RDS can also provide a highly available MySQL database by replication.
- *Elastic File System (EFS)*—WordPress itself consists of PHP and other application files. User uploads, for example images added to an article, are stored as files as well. By using a network file system, your virtual machines can access these files. EFS provides a scalable, highly available, and durable network filesystem using the NFSv4.1 protocol.

- *Security groups*— Control incoming and outgoing traffic to your virtual machine, your database, or your load balancer with a firewall. For example, use a security group allowing incoming HTTP traffic from the internet to port 80 of the load balancer. Or restrict network access to your database on port 3306 to the virtual machines running your web servers.

Figure 2.1 shows all the parts of the infrastructure in action. Sounds like a lot of stuff to set up, so let's get started!

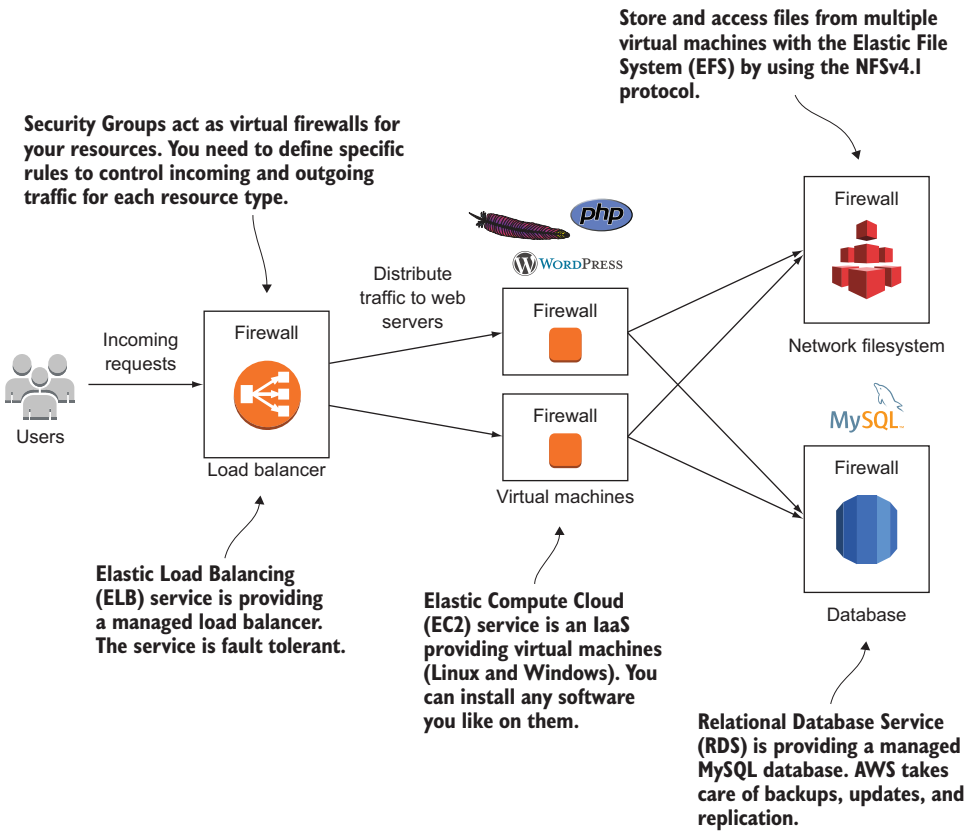

Figure 2.1 The company's blogging infrastructure consists of two load-balanced web servers running WordPress, a network filesystem, and a MySQL database server.

If you expect pages of instructions, you'll be happy to know that you can create all that with just a few clicks. Doing so is possible by using a service called AWS CloudFormation that you will learn about in detail in chapter 4. AWS CloudFormation will do all of the following automatically in the background:

1. Create a load balancer (ELB).
2. Create a MySQL database (RDS).

3. Create a network filesystem (EFS).
4. Create and attach firewall rules (security groups).
5. Create two virtual machines running web servers:
   a. Create two virtual machines (EC2).
   b. Mount the network filesystem.
   c. Install Apache and PHP.
   d. Download and extract the 4.8 release of WordPress.
   e. Configure WordPress to use the created MySQL database (RDS).
   f. Start the Apache web server.

To create the infrastructure for your proof-of-concept, open the AWS Management Console at https://console.aws.amazon.com. Click Services in the navigation bar, and select the CloudFormation service. You can use the search function to find CloudFormation more easily. You'll see a page like the one shown in figure 2.2.

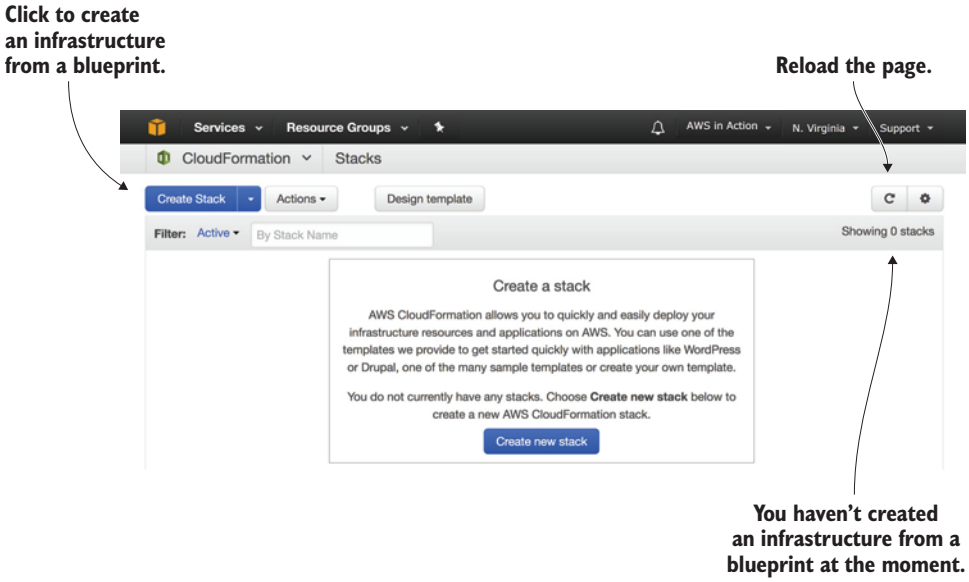

Figure 2.2 Overview of CloudFormation

DEFAULT REGION FOR EXAMPLES All examples in this book use N. Virginia (also called us-east-1) as the default region. Exceptions are indicated. Please make sure you switch to the region N. Virginia before starting to work on an example. When using the AWS Management Console, you can check and switch the region on the right side of the main navigation bar at the top.

Click Create Stack to start the four-step wizard, as shown in figure 2.3.

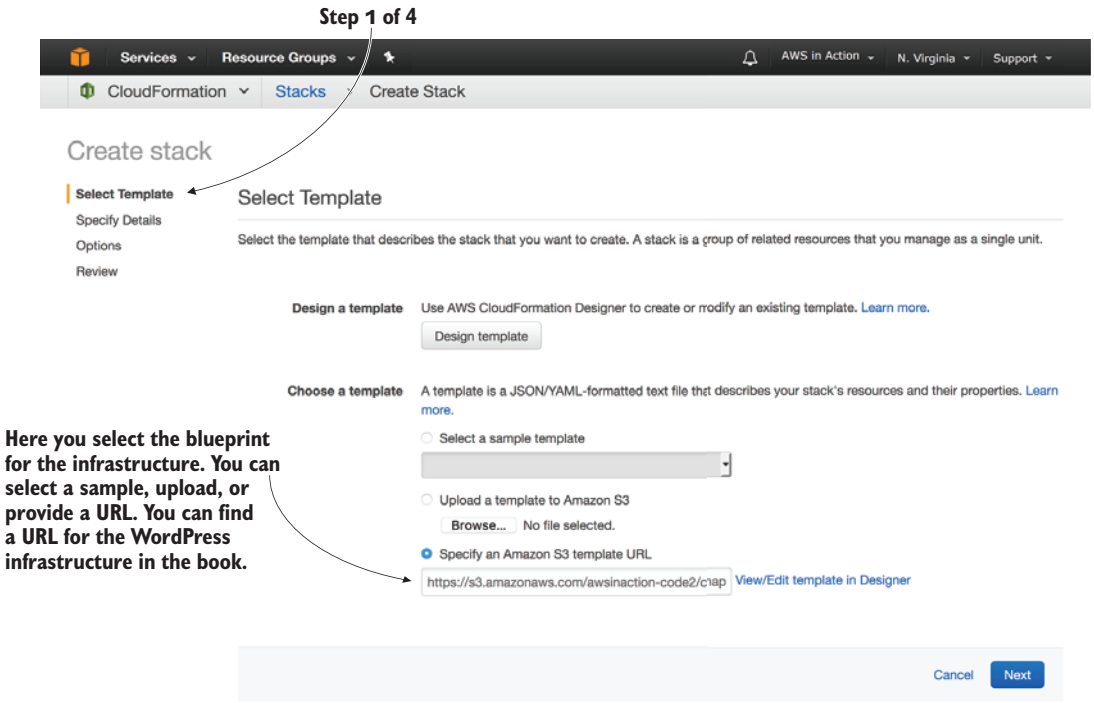

Figure 2.3 Creating the stack for the proof-of-concept: step 1 of 4

Choose Specify an Amazon S3 template URL and enter `https://s3.amazonaws.com/awsinaction-code2/chapter02/template.yaml` to use the template prepared for this chapter. Proceed with the next step of the wizard.

Specify `wordpress` as the Stack name and set the `KeyName` to `mykey` in the Parameters section as shown in figure 2.4.

The next step is to specify tags for your infrastructure, as illustrated by figure 2.5. A *tag* consists of a key and a value, and can be used to add metadata to all parts of your infrastructure. You can use tags to differentiate between testing and production resources, add a cost center to easily track costs in your organization, or mark resources that belong to a certain application if you host multiple applications in the same AWS account.

Figure 2.5 shows how to configure the tag. In this example, you'll use a tag to mark all resources that belong to the `wordpress` system. This will help you to easily find all the parts of your infrastructure later. Use a custom tag consisting of the key `system` and the value `wordpress`. Afterward, press the Next button to proceed to the next step. You can define your own tags as long as the key name is shorter than 128 characters and the value has fewer than 256 characters.

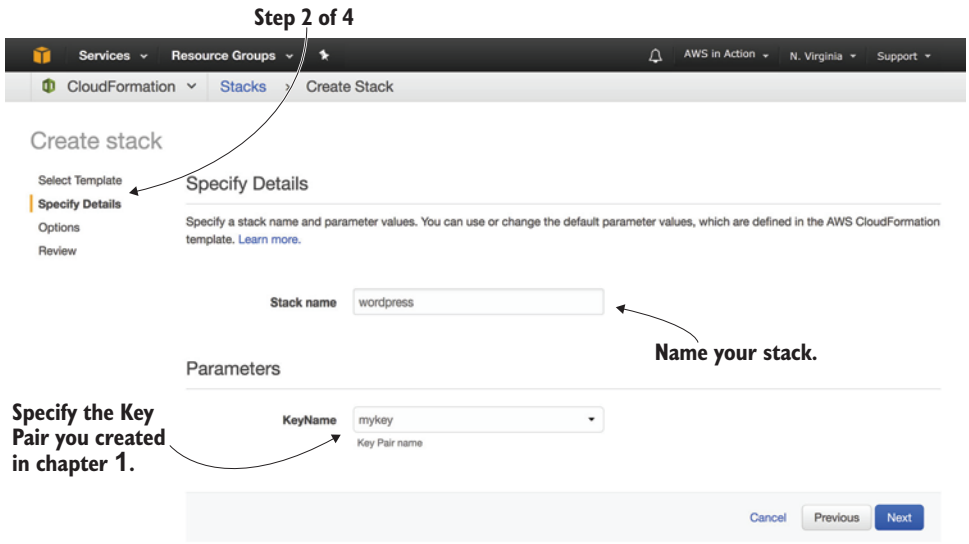

Figure 2.4 Creating the stack for the proof-of-concept: step 2 of 4

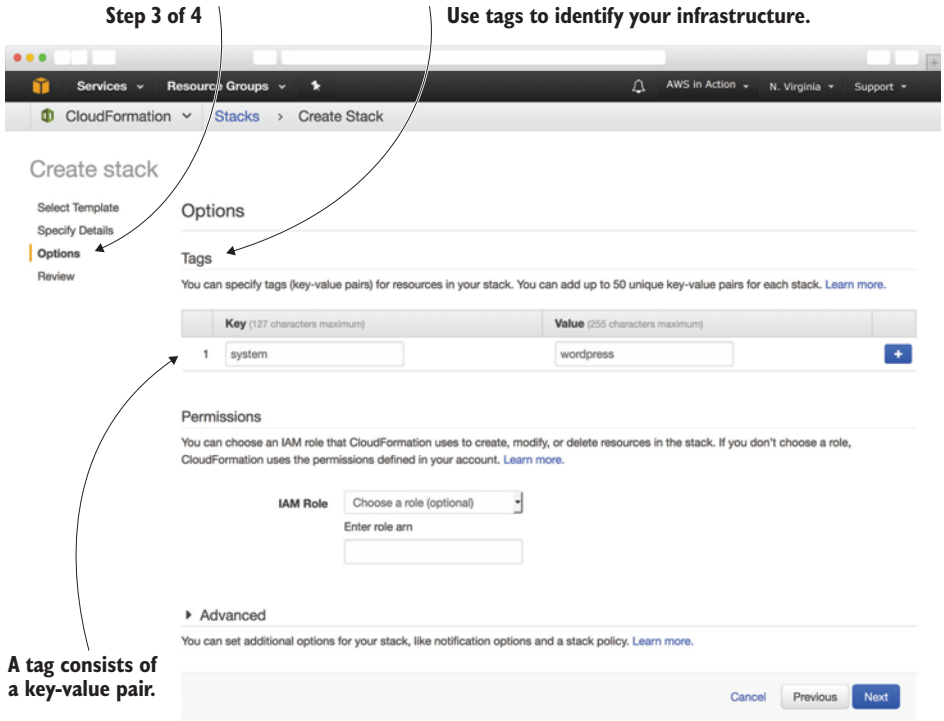

Figure 2.5 Creating the stack for the proof-of-concept: step 3 of 4

Additional CloudFormation Stack options

It is possible to define specific permissions used to manage resources, as well as to set up notifications and other advanced options. You won't need these options for 99% of the use cases, so we don't cover them in our book. Have a look at the CloudFormation User Guide (http://mng.bz/njoZ) if you're interested in the details.

Figure 2.6 shows the confirmation page. Please click on the Estimate cost link to open a cost estimation of your cloud infrastructure in a new browser tab. Don't worry, the example is covered by the Free Tier and you'll learn about all the details of the cost estimation in section 2.3. Switch back to the original browser tab, and click Create.

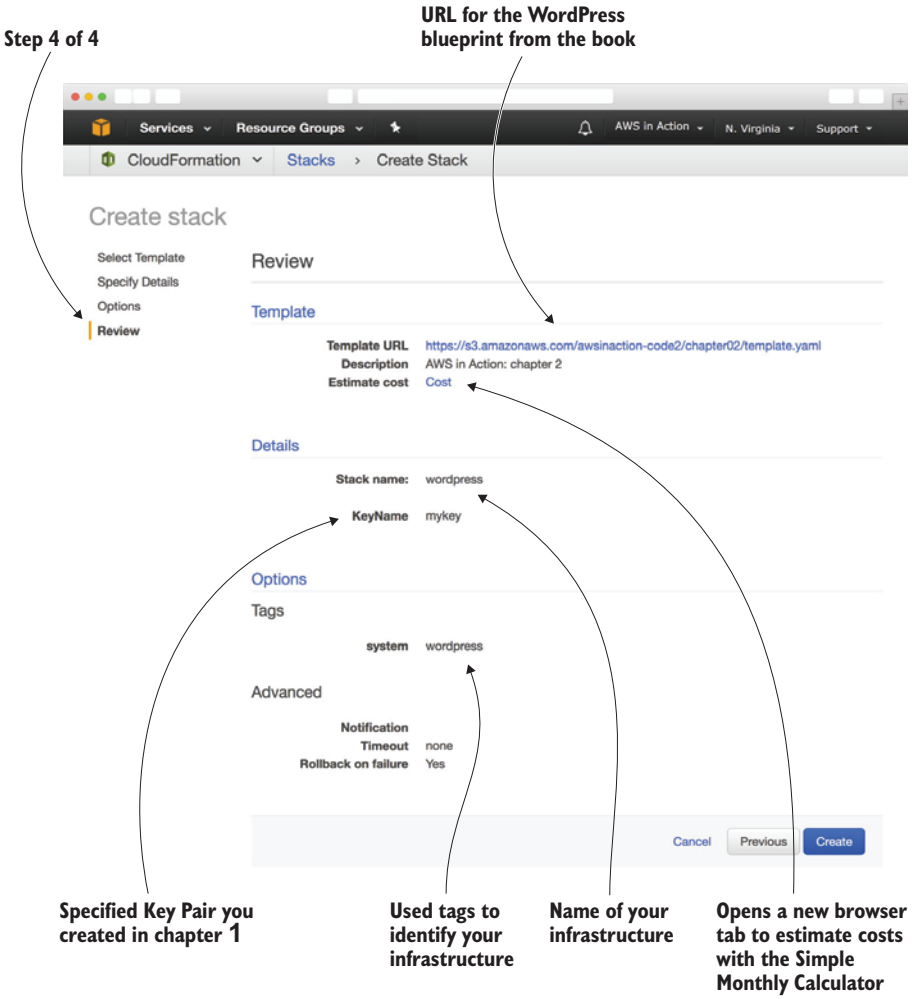

Figure 2.6 Creating the stack for the proof-of-concept: step 4 of 4

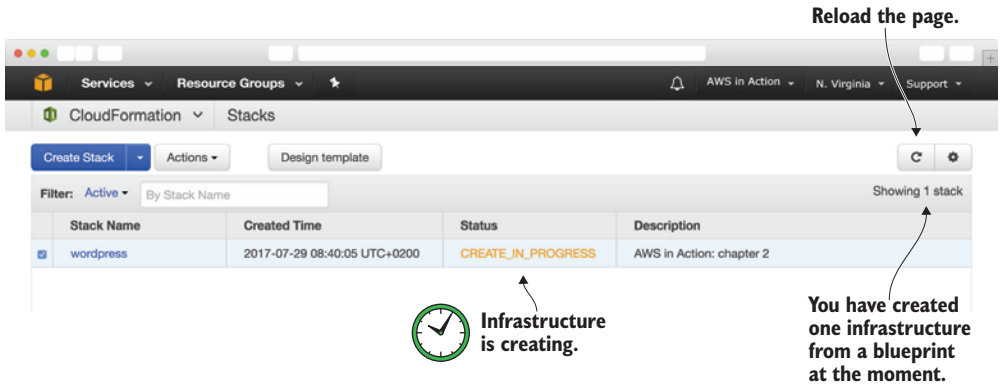

Figure 2.7 CloudFormation is creating the resources needed for WordPress.

Your infrastructure will now be created. Figure 2.7 shows that `wordpress` is in state `CREATE_IN_PROGRESS`. It's a good time to take a break; come back in five minutes, and you'll be surprised.

After all the needed resources have been created, the status will change to `CREATE_COMPLETE`. Be patient and hit the refresh icon from time to time if your status still shows as `CREATE_IN_PROGRESS`.

Select the check box at the beginning of the row containing your wordpress stack. Switch to the Outputs tab, as shown in figure 2.8. There you'll find the URL to your WordPress installation; click the link to open it in your browser.

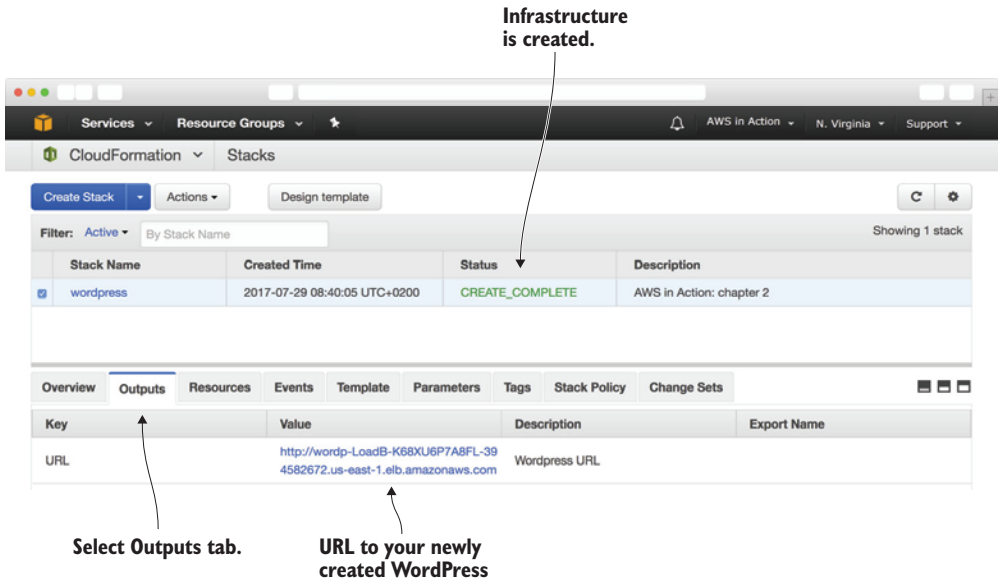

Figure 2.8 Blogging infrastructure has been created successfully.

You may ask yourself, how does this work? The answer is *automation*.

> **Automation references**
>
> One of the key concepts of AWS is automation. You can automate everything. In the background, your blogging infrastructure was created based on a blueprint. You'll learn more about blueprints and the concept of programming your infrastructure in chapter 4. You'll learn to automate the installation of software in chapter 5.

You'll explore the blogging infrastructure in the next section to get a better understanding of the services you're using.

2.2 Exploring your infrastructure

Now that you've created your blogging infrastructure, let's take a closer look at it. Your infrastructure consists of the following:

- Web servers running on virtual machines
- Load balancer
- MySQL database
- Network filesystem

You'll use the Management Console's resource groups feature to get an overview.

2.2.1 Resource groups

A *resource group* is a collection of AWS resources. *Resource* in AWS is an abstract term for something like an virtual machine, a security group, or a database. Resources can be tagged with key-value pairs, and resource groups specify which tags are needed for a resource to belong to the group. Furthermore, a resource group specifies the region(s) the resource must reside in. You can use resource groups for grouping resources if you run multiple systems in the same AWS account.

Remember that you tagged the blogging infrastructure with the key `system` and the value `wordpress`. From now on, we'll use this notation for key-value pairs: `(system:wordpress)`. You'll use that tag to create a resource group for your WordPress infrastructure. Click Create a Resource Group as shown in figure 2.9.

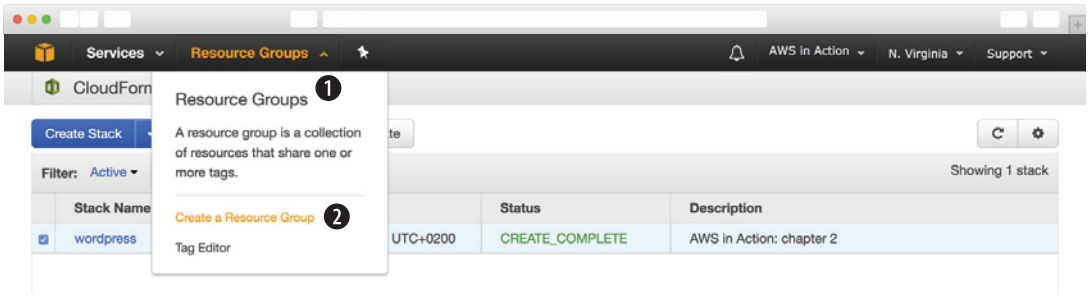

Figure 2.9 Open resource groups

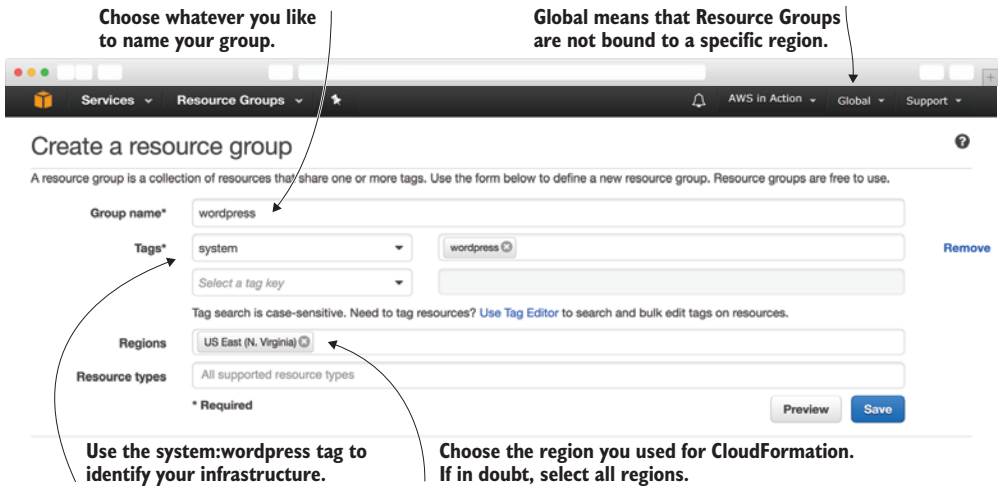

Figure 2.10 Creating a resource group for your blogging infrastructure

You'll now create a resource group as illustrated in 2.10:

1. Set Group Name to `wordpress` or whatever you like.
2. Add the tag `system` with the value `wordpress`.
3. Select the region you're in, which is probably N. Virginia. If you don't know the region you're in, select All Regions.
4. Click Save.

2.2.2 Virtual machines

Now you'll see the screen shown in figure 2.11. Select Instances under EC2 on the left to see your virtual machines. By clicking the arrow icon in the Go column, you can easily jump to the details of a single virtual machine.

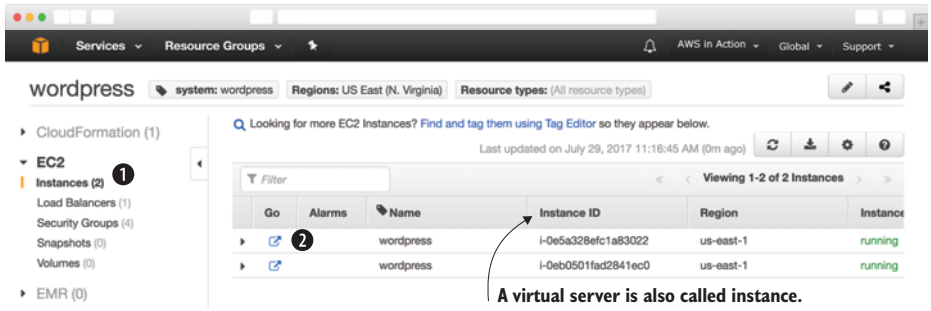

Figure 2.11 Blogging infrastructure virtual machines via resource groups

You're now looking at the details of your virtual machine, also called an EC2 instance. Figure 2.12 shows an extract of what you see. The interesting details are as follows:

- *Instance type*—Tells you about how powerful your EC2 instance is. You'll learn more about instance types in chapter 3.
- *IPv4 Public IP*—The IP address that is reachable over the internet. You can use that IP address to connect to the virtual machine via SSH.
- *Security groups*—If you click View Rules, you'll see the active firewall rules, like the one that enables port 22 from all sources (0.0.0.0/0)
- *AMI ID*—Remember that you used the Amazon Linux OS. If you click the AMI ID, you'll see the version number of the OS, among other things.

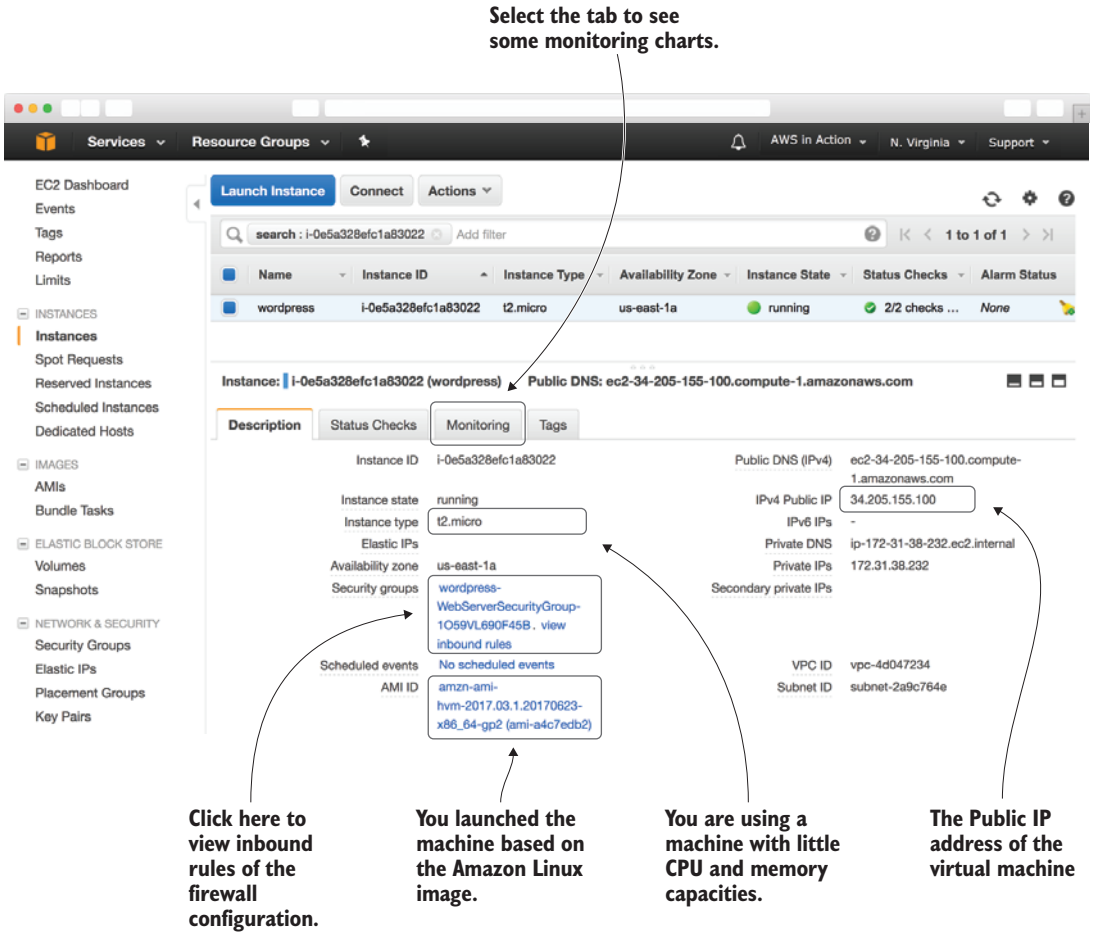

Figure 2.12 Details of web servers running the blogging infrastructure

Select the Monitoring tab to see how your virtual machine is utilized. This tab is essential if you really want to know how your infrastructure is doing. AWS collects some metrics and shows them here. For example, if the CPU is utilized more than 80%, it might be a good time to add another virtual machine to prevent increasing response times. You will learn more about monitoring virtual machines in section 3.2.

2.2.3 Load balancer

AWS released a new load balancer type, called Application Load Balancer, in August 2016. Unfortunately, our resource group does not list Application Load Balancers yet. Therefore, click Load Balancers in the sub navigation of the EC2 service as shown in figure 2.13.

Select your load balancer from the list to show more details. Your internet-facing load balancer is accessible from the internet via an automatically generated DNS name.

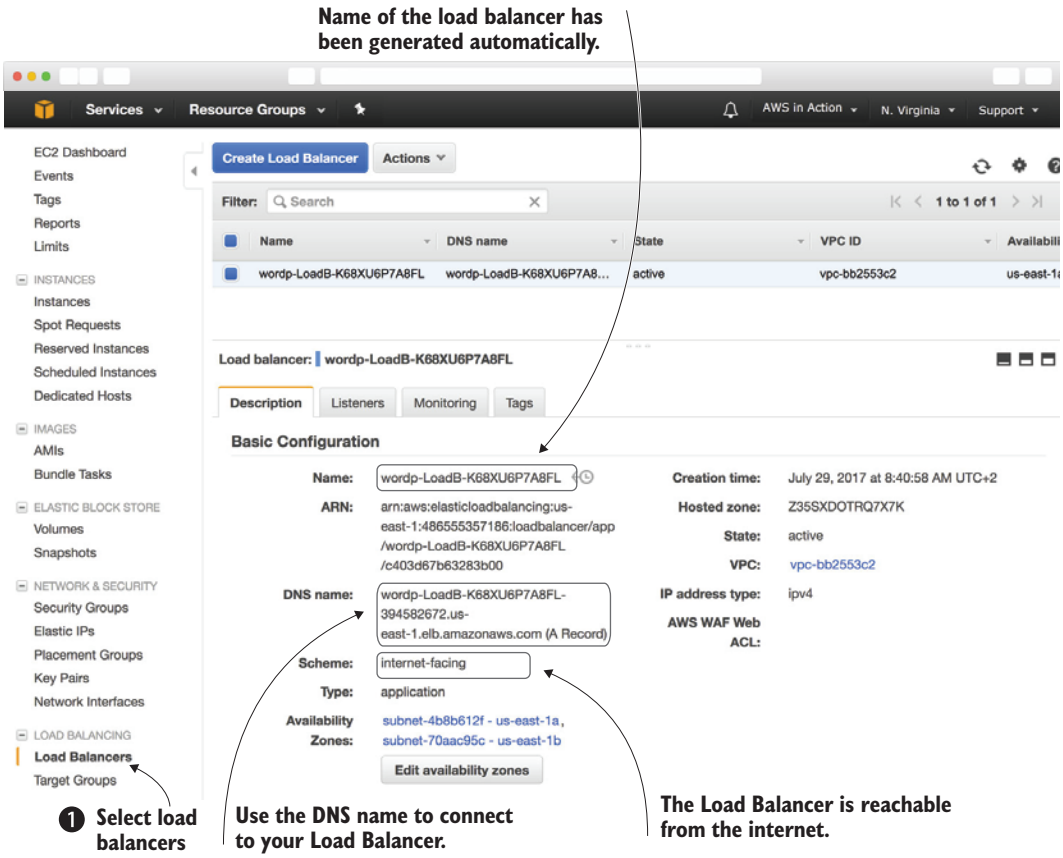

Figure 2.13 Get details about your load balancer.

The load balancer forwards incoming requests to one of your virtual machines. A target group is used to define the targets for a load balancer. You'll find your target group after switching to Target Groups through the sub navigation of the EC2 service as shown in figure 2.14.

The load balancer performs health checks to ensure requests are routed to healthy targets only. Two virtual machines are listed as targets for the target group. As you can see in the figure, the status of both virtual machines is healthy.

As before, there is a Monitoring tab where you can find interesting metrics that you should watch in production. If the traffic pattern changes suddenly, this indicates a potential problem with your system. You'll also find metrics indicating the number of HTTP errors, which will help you to monitor and debug your system.

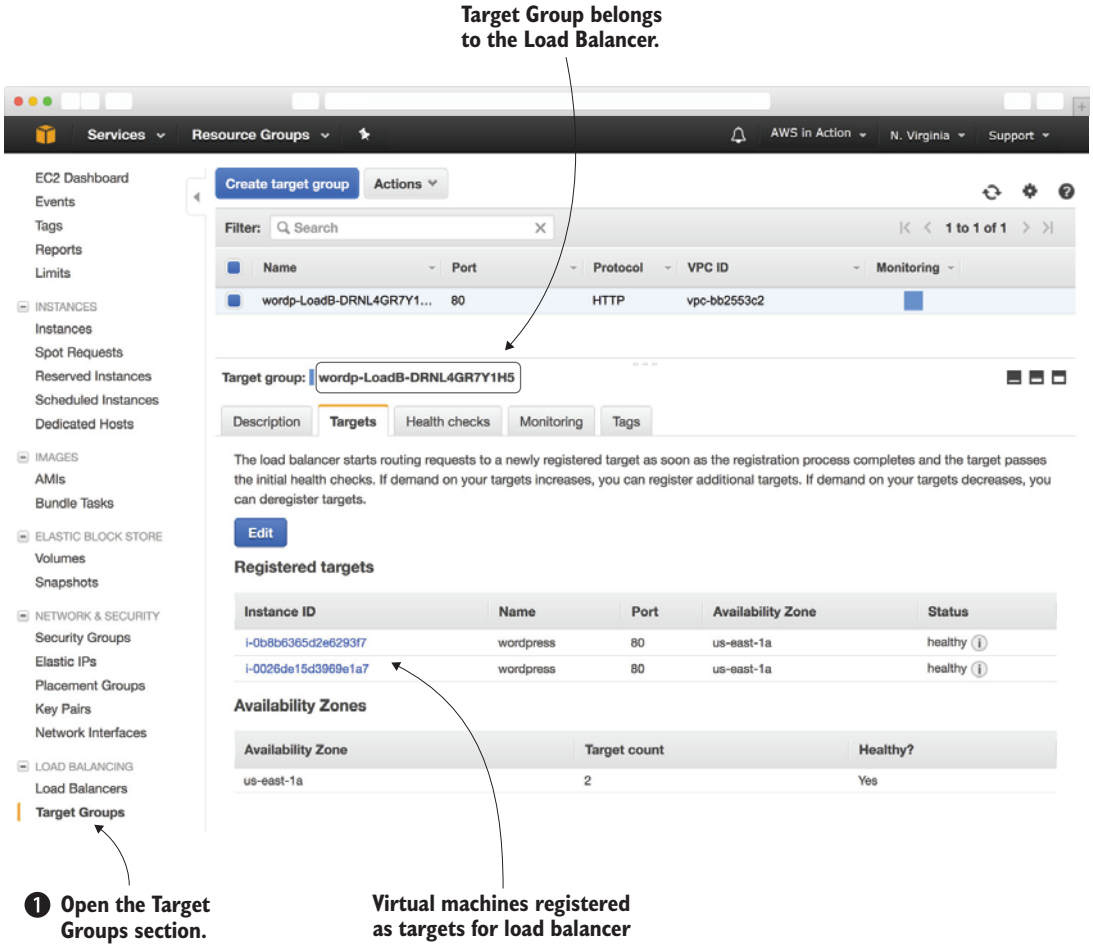

Figure 2.14 Details of target group belonging to the load balancer

2.2.4 MySQL database

The MySQL database is an important part of your infrastucture; you'll look at it next. Go back to the resource group named `wordpress`. Select DB Instances under RDS on the left. By clicking the arrow icon in the Go column, as shown in figure 2.15, you can easily jump to the details of the database.

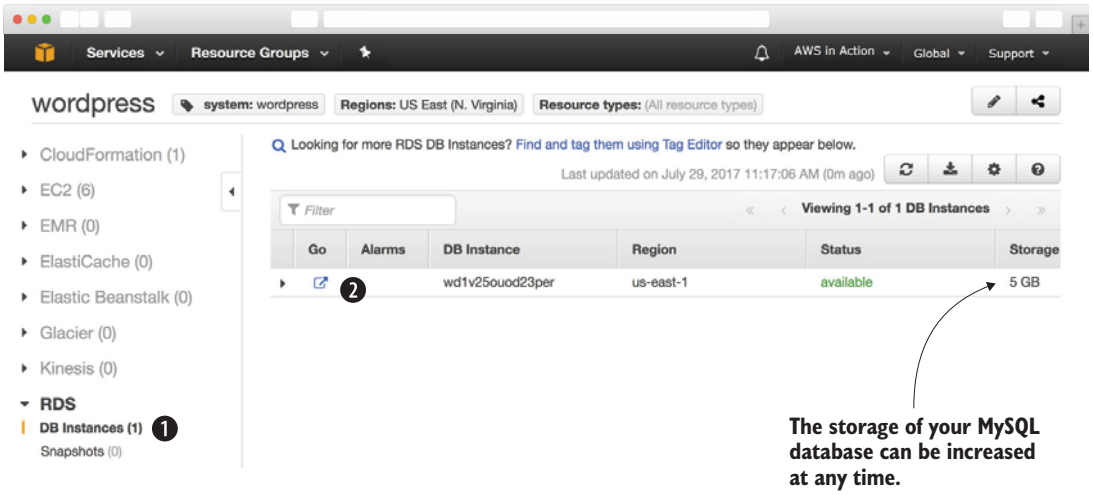

Figure 2.15 Blogging infrastructure MySQL via resource groups

The details of your MySQL database are shown in figure 2.16. The RDS offers SQL databases as managed services, complete with backups, patch management, and high availability. As shown in figure 2.16 automated backups are disabled, as they are not needed for our proof-of-concept without any critical data. You can also find the maintenance window used by AWS to apply patches automatically in the Details section.

WordPress requires a MySQL database, so you have launched a database instance with the MySQL engine as noted in figure 2.16. Your blog receives a low amount of traffic, so the database doesn't need to be very powerful. A small instance class with a single virtual CPU and 1 GB memory is sufficient. Instead of using SSD storage, you are using magnetic disks, which is cheaper and sufficient for a web application with around 1,000 visitors per day.

As you'll see in chapter 9, other database engines, such as PostgreSQL or Oracle Database, are available as well as more powerful instance classes, offering up to 32 cores with 244 GB memory.

Common web applications use a database to store and query data. That is true for WordPress as well. The Content Management System (CMS) stores blog posts, comments, and more within a MySQL database.

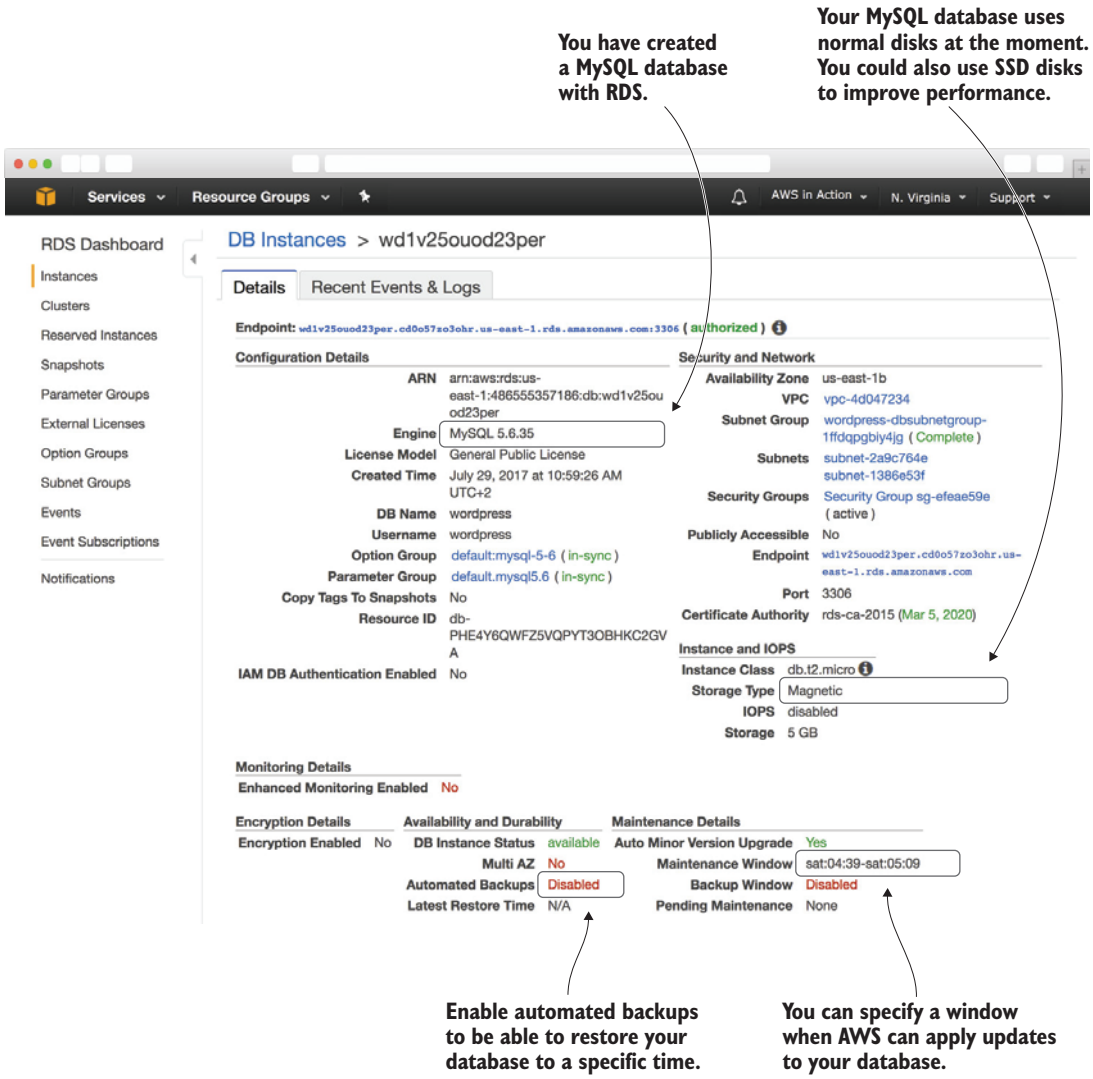

Figure 2.16 Details of the MySQL database storing data for the blogging infrastructure

But WordPress also stores data outside the database on disk. For example, if an author uploads an image for their blog post, the file is stored on disk. The same is true when you are installing plug-ins and themes as an administrator.

2.2.5 Network filesystem

The EFS is used to store files and access them from multiple virtual machines. EFS is a storage service accessible through the NFS protocol. To keep things simple, all files that belong to WordPress are stored on EFS so they can be accessed from all virtual machines. This includes PHP, HTML, CSS, and PNG.

EFS is not available from your resource group, unfortunately. Select EFS from the service menu to get more information about your NFS, as shown in figure 2.17. You can find the name, DNS name, and the mount targets of the filesystem in the Details section.

To mount the Elastic File System from a virtual machine, mount targets are needed. You should use two mount targets for fault tolerance. The network filesystem is accessible using a DNS name for the virtual machines.

Now it's time to evaluate costs. You'll analyze the costs of your blogging infrastructure in the next section.

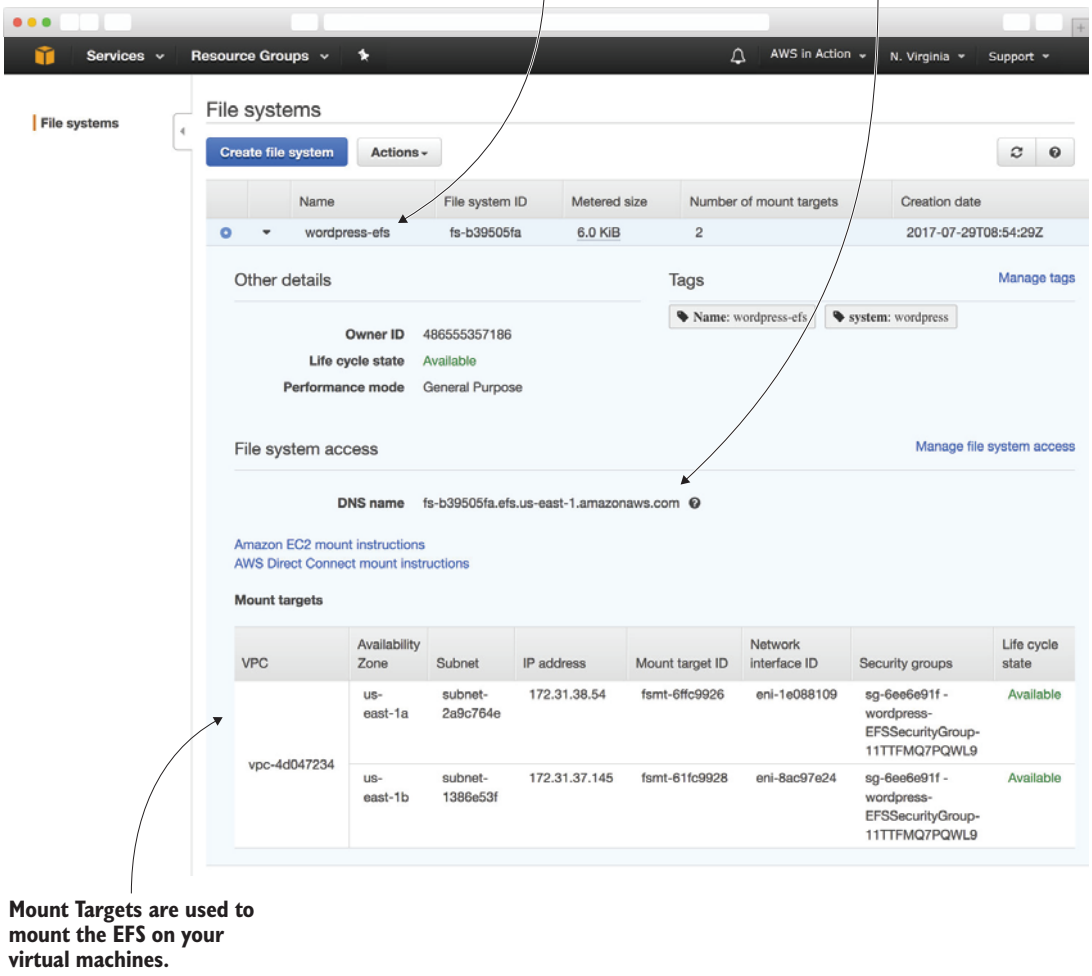

Figure 2.17 NFS used to store the WordPress application and user uploads

2.3 How much does it cost?

Part of evaluating AWS is estimating costs. To analyze the cost of your blogging infrastructure, you'll use the AWS Simple Monthly Calculator. Remember that you clicked the Cost link in the previous section to open a new browser tab. Switch to that tab, and you'll see a screen like that in figure 2.18. If you closed the tab, go to http://mng.bz/x6A0 instead. Click Estimate of your Monthly Bill, and expand the rows marked Amazon EC2 Service and Amazon RDS Service.

In this example, your infrastructure will cost around $35 USD per month. Prices for some services vary per region. That's why the estimation of your monthly bill may be different if you choose another region than N. Virginia (us-east-1).

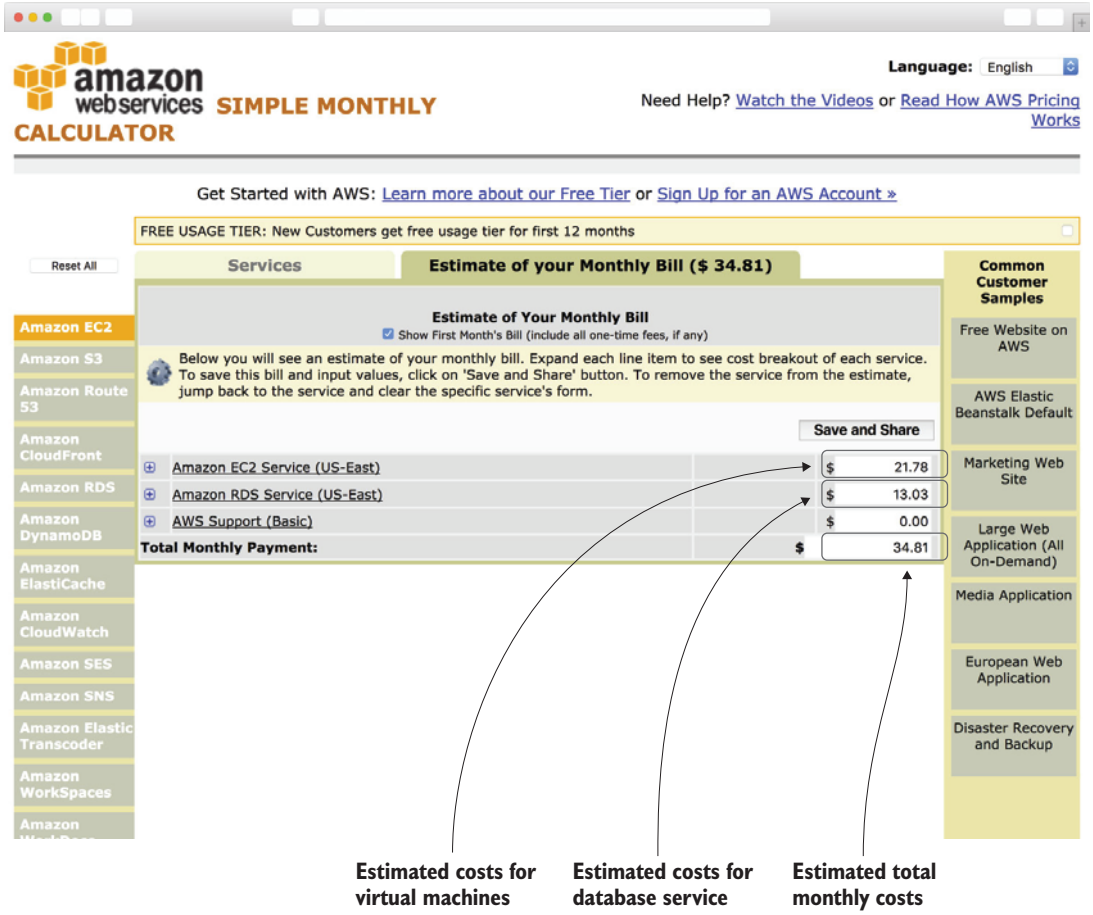

Figure 2.18 Blogging infrastructure cost calculation

Unfortunately the somewhat-new Application Load Balancer is not included in the estimation yet. The estimation is also missing some other details. Table 2.1 shows a more accurate cost calculation.

Table 2.1 More detailed cost calculation for blogging infrastructure

AWS service	Infrastructure	Pricing	Monthly cost
EC2	Virtual machines	2 * 732.5 hours * $0.012 (t2.micro) 2 * $2.10 (detailed monitoring)	$21.78
EC2	Storage	2 * 8 GB * $0.10 per month	$1.60
Application Load Balancer	Load balancer	732.5 hours * $0.0225 (load balancer hour) 732.5 hours * $0.008 (load balancer capacity unit)	$22.34
Application Load Balancer	Outgoing traffic	1 GB * $0.00 (first GB) 99 GB * $0.09 (up to 10 TB)	$8.91
RDS	MySQL database instance	732.5 hours * $0.017	$12.45
RDS	Storage	5 GB * $0.115	$0.58
EFS	Storage	5 GB * $0.3	$1.50
			$69.16

Keep in mind that this is only an estimate. You're billed based on actual use at the end of the month. Everything is on-demand and usually billed by seconds or gigabyte of usage. But what factors might influence how much you actually use this infrastructure?

- *Traffic processed by the load balancer*—Expect costs to go down in December and in the summer when people are on vacation and not looking at your blog.
- *Storage needed for the database*—If your company increases the amount of content in your blog, the database will grow, so the cost of storage will increase.
- *Storage needed on the NFS*—User uploads, plug-ins, and themes increase the amount of storage needed on the NFS, which will also increase costs.
- *Number of virtual machines needed*—Virtual machines are billed by seconds of usage. If two virtual machines aren't enough to handle all the traffic during the day, you may need a third machine. In that case, you'll consume more seconds of virtual machines.

Estimating the cost of your infrastructure is a complicated task, but that is also true if your infrastructure doesn't run in AWS. The benefit of using AWS is that it's flexible. If your estimated number of virtual machines is too high, you can get rid of a machine and stop paying for it. You will learn more about the pricing model of the different AWS services during the course of this book.

You have completed the proof-of-concept for migrating your company's blog to AWS. It's time to shut down the infrastructure and complete your migration evaluation.

2.4 Deleting your infrastructure

Your evaluation has confirmed that you can migrate the infrastructure needed for the company's blog to AWS from a technical standpoint. You have estimated that a load balancer, virtual machines, MySQL database, as well as a NFS capable of serving 1,000 people visiting the blog per day will cost you around $70 USD per month on AWS. That is all you need to come to a decision.

Because the infrastructure does not contain any important data and you have finished your evaluation, you can delete all the resources and stop paying for them.

Go to the CloudFormation service in the Management Console, and take the following steps in figure 2.19:

1. Select the check box at the beginning of the row containing your wordpress stack.
2. Open the Actions menu by clicking Actions.
3. Click Delete Stack.

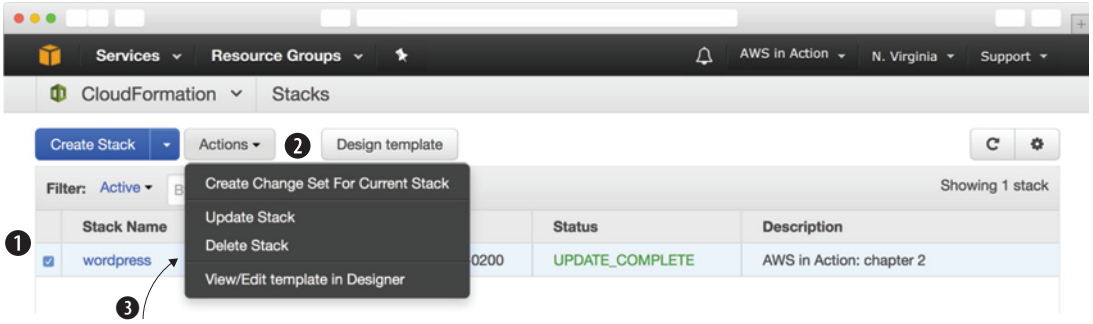

Figure 2.19 Delete your blogging infrastructure.

After you confirm the deletion of the infrastructure, as shown in figure 2.20, it takes a few minutes for AWS to delete all of the infrastructure's dependencies.

This is an efficient way to manage your infrastructure. Just as the infrastructure's creation was automated, its deletion is also. You can create and delete infrastructure on-demand whenever you like. You only pay for infrastructure when you create and run it.

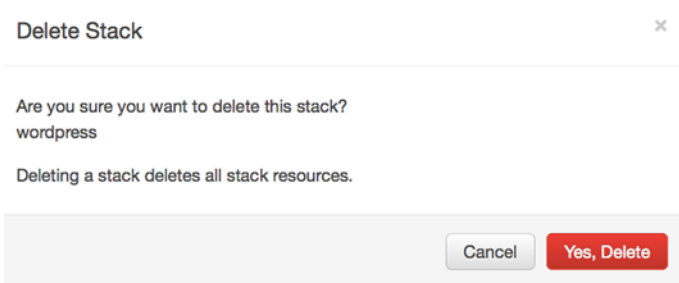

Figure 2.20 Confirming deletion of your blogging infrastructure

Summary

- Creating a blogging infrastructure can be fully automated.
- Infrastructure can be created at any time on-demand, without any up-front commitment for how long you'll use it.
- You pay for your infrastructure based on usage. For example, you are paying for a virtual machine per second of usage.
- Infrastructure consists of several parts, such as virtual machines, load balancers, and databases.
- Infrastructure can be deleted with one click. The process is powered by automation.